

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-215173

(43)Date of publication of application : 04.08.2000

(51)Int.Cl.

G06F 15/00
G06F 12/00
G06F 15/16
// G06F 17/30

(21)Application number : 11-013394

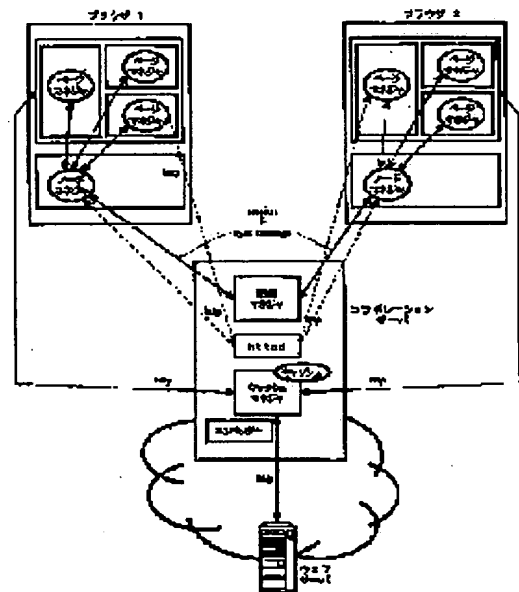
(71)Applicant : INTERNATL BUSINESS MACH
CORP <IBM>

(22)Date of filing : 21.01.1999

(72)Inventor : KOBAYASHI MAKOTO
SHINOZAKI MASAHIDE
SAKAIRI TAKASHI**(54) BROWSER SHARING METHOD AND SYSTEM****(57)Abstract:**

PROBLEM TO BE SOLVED: To provide the browser sharing method and system which is not required to previously install a control mechanism in a computer to perform the sharing of a browser.

SOLUTION: A collaboration server is prepared to accumulate the sharing pages from a Web server holding the original pages and includes a cache manager which accumulates the pages generated dynamically by the Web server for the sharing purpose, a communication manager which performs the session management such as the communication and the participation/withdrawal between the node managers which manage browsers existing on every user machine, an ordinary Web server which downloads a mechanism to attain the sharing of the said Web server and an embedder which embeds a page manager in every page. Each of plural user machines consists of an existing Web browser which can execute both Java and Script and a page manager which is embedded in every page.

**LEGAL STATUS**

[Date of request for examination] 21.12.1999

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number] 3437933

[Date of registration] 06.06.2003

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's]

decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-215173

(P2000-215173A)

(43) 公開日 平成12年8月4日(2000.8.4)

(51) Int.Cl. ⁷	識別記号	F I	テーマコード*(参考)
G 0 6 F 15/00	3 9 0	G 0 6 F 15/00	3 9 0 5 B 0 4 5
12/00	5 4 7	12/00	5 4 7 H 5 B 0 7 5
15/16	6 2 0	15/16	6 2 0 G 5 B 0 8 2
// G 0 6 F 17/30		15/40	3 1 0 C 5 B 0 8 5

審査請求 有 請求項の数11 O L (全 10 頁)

(21) 出願番号 特願平11-13394

(22) 出願日 平成11年1月21日(1999.1.21)

(71) 出願人 390009531

インターナショナル・ビジネス・マシー
ズ・コーポレーション

INTERNATIONAL BUSIN
ESS MACHINES CORPO
RATION

アメリカ合衆国10504、ニューヨーク州
アーモンク (番地なし)

(74) 代理人 100086243

弁理士 坂口 博 (外1名)

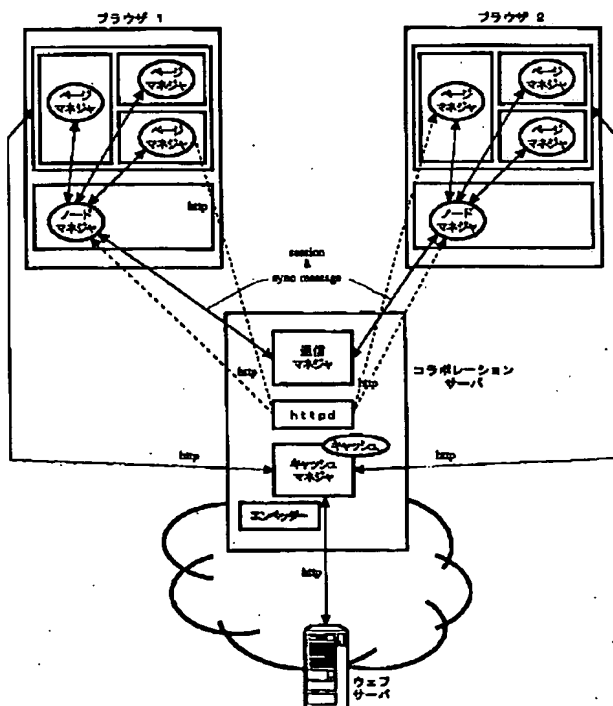
最終頁に続く

(54) 【発明の名称】 ブラウザ共有方法及びシステム

(57) 【要約】 (修正有)

【課題】 共有を行う制御機構をあらかじめコンピュータにインストールする必要がない、ブラウザ共有方法及びシステムを提供する。

【解決手段】 本来のページを保持するウェブサーバから共有用のページを蓄積するためのコラボレーションサーバを用意する。コラボレーションサーバはもとのウェブサーバで動的に生成されるページを共有用に蓄積するキャッシュマネージャと、各ユーザマシン上のブラウザを管理するノードマネージャ間の通信と参加・退出などのセッション管理をするコミュニケーションマネージャと、ウェブサーバ共有を実現する機構をダウンロードするための通常のウェブサーバと、各ページにページを管理するページマネージャを埋め込むエンベッダーからなる。共有が行われる複数のユーザマシンは、Java および Script を実行することのできる既存のウェブブラウザと、各ページに埋め込まれたページマネージャから構成される。



【特許請求の範囲】

【請求項 1】サーバと、複数のコンピュータとからなるブラウザ共有システムであって、前記複数のコンピュータが、ページを閲覧するブラウザと、ページを管理するページマネージャと、ページマネージャとサーバとの間の通信を行いブラウザを管理するノードマネージャとを含み、前記ページマネージャが、自ページの変化を検出し、その変化をノードマネージャに送信し、ノードマネージャが、前記変化をサーバに送信するか、若しくは他のコンピュータのページの変化をノードマネージャから受信し、その変化を自ページに反映させる手段を有する、ブラウザ共有システム。

【請求項 2】前記サーバが、ページを蓄積するキャッシュマネージャと、前記複数のコンピュータとの間のセッション管理をするコミュニケーションマネージャと、各ページにページを管理するページマネージャを埋め込むエンベッダーとを含む、請求項 1 記載のシステム。

【請求項 3】前記ページマネージャが、ページコントローラおよびページコミュニケーションータからなり、ページコントローラが、ページ要素の変化を検出して、該変化をページコミュニケーションータ経由でノードマネージャに送信するか、またはページコントローラが、他のコンピュータのページの変化をページコミュニケーションータ経由でノードマネージャから受信し、受け取った変化を自ページの要素に反映させる、請求項 1 記載のシステム。

【請求項 4】前記ページ要素の変化が、ページの移動の変化、テキストやボタンを含むフォーム要素の値の変化、ページのスクロール位置の変化、またはリモートポインタの操作である、請求項 3 記載のシステム。

【請求項 5】前記ページマネージャが、ページのフレーム階層構造を解析し、この解析結果に基づき対応するページマネージャと通信を行う、請求項 1 記載のシステム。

【請求項 6】前記ノードマネージャが、共有されたブラウザ内のページとは独立の移動しないページ内に常駐し、ページの移動等に伴い、動的に生成・消滅するページマネージャ間の通信を制御する、請求項 1 記載のシステム。

【請求項 7】前記ノードマネージャが、ページの遷移ヒストリーを含むページ情報を管理する、請求項 1 記載のシステム。

【請求項 8】ページマネージャとノードマネージャは、同一のドメインを持つ Java アプレットとして埋め込まれ、ページマネージャとノードマネージャの間で共有メモリによるデータ通信を行なう、請求項 1 記載のシステム。

【請求項 9】複数のコンピュータ間においてブラウザ共有を行なうためのサーバであって、コンピュータからブラウザ共有の信号を受信する手段と、コンピュータにブラウザ管理を行なうノードマネージャを送信する手段と、コンピュータからブラウザによるページ閲覧要求を受信する手段と、前記ページ閲覧要求に従い、ページ管理を行なうページマネージャを埋め込んだ、要求ページをコ

ンピュータに送信する手段と、ページマネージャがノードマネージャを介して送信した、ページ変化情報を受信する手段と、前記ページ変化情報を他のコンピュータに送信する手段と、を有する、サーバ。

【請求項 10】ブラウザを複数のコンピュータ間で共有する方法であって、コンピュータのブラウザ起動に伴い、サーバから、ノードマネージャをコンピュータにロードするステップと、サーバとノードマネージャとの間で通信を確立するステップとノードマネージャが、共有メモリを割り当てるステップと、ブラウザによるページ閲覧に伴い、サーバにおいて、要求されたページにページマネージャを埋め込むステップと、前記共有メモリを介して、ノードマネージャとページマネージャとの間で通信を確立するステップと、ページ閲覧に伴う、ページの変化を前記共有メモリを介してノードマネージャに送信するか、または他のコンピュータのページ変化を前記共有メモリを介して、ノードマネージャから受信し、該変化を次ページに反映するステップと、を有する、ブラウザ共有方法。

【請求項 11】ブラウザを複数のコンピュータ間で共有させるためのプログラムを含む媒体であって、該プログラムが、前記コンピュータに、サーバとの間で通信を確立する機能と、共有メモリを割り当てる機能と、ブラウザによるページ閲覧に伴い、サーバにページ要求を出す機能と、サーバからページを管理するページマネージャの埋め込まれたページを受信する機能と、ページマネージャから受け取ったページ変化を前記共有メモリを介して、サーバに送信するか、または他のコンピュータのページ変化をサーバから受信し、該変化をページマネージャに前記共有メモリを介して送信する機能と、を実現させるプログラムを含む媒体。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、ブラウザ共有方法及びシステムに関し、特にページ自体にページ内の各要素を制御する機構を埋め込むことにより、ユーザ側でのインストールなどの手間なく、既存のウェブブラウザと既存のウェブページに対して高機能なりリアルタイムブラウザ共有を実現する方法およびシステムに関する発明である。

【0002】

【従来の技術】既存のウェブブラウザを用いて、標準のウェブページをそのままの形式で共有できる方式に関しては、種々知られている。たとえば特願平 9-254997（社内番号 Y09-96-120）などが知られている。しかしながら、このような従来技術では、すべてのユーザのところで共有化システムのインストールがあらかじめ必要となる。特願平 9-254997 では、共有のためのデータ交換は、ブラウザ自身が備えているアプリケーションインターフェイス（具体的にはページの移動の通知イベント、現ページの問い合わせ、および

3

ページの設定機能)、およびオペレーティングシステムレベルでのインターフェイス(具体的には、ブラウザーのウィンドウがもつメッセージキューに対するGETおよびSETの機能)、という2つのインターフェイスを通してブラウザー状態の同期を実現している。このとき、アプリケーションインターフェイスおよびメッセージキューインターフェイスはいずれもブラウザーアプリケーションプロセスの外側からのみアクセスが可能であり、したがって共有を実現するためのモジュールはブラウザーの外にあるために、アプレットとして自動ダウンロードができる形式ではなくあらかじめインストールしておく必要がある。またブラウザーが動くOSやウィンドウシステムに依存するので、プラットフォームごとの実装が必要になるという問題も抱えていた。

【0003】図1に特願平9-254997における、ブラウザ共有の仕組みを示す。ウェブサーバのページをコラボレーションサーバ上で共有するために、顧客とコールセンター・オペレータにはブラウザ以外に、共有のためのプログラム(WebShare)を予めインストールしておく必要がある。この共有プログラムはブラウザのAPIとイベントをフックして共有を可能にしている。またこのようなプログラムはブラウザーが動くOSやウィンドウシステムに依存するので、プラットフォームごとの実装が必要になる。

【0004】

【発明が解決しようとする課題】本発明は上記従来技術の問題点に鑑み発明されたもので、本発明が解決しようとする課題は、共有を行う制御機構をあらかじめコンピュータにインストールする必要がない、ブラウザ共有方法及びシステムを提供することである。また別の課題は、OSやウィンドウシステムの機能に依存しない、多様なプラットフォームで動作する、ブラウザ共有方法及びシステムを提供することである。また別の課題は、アプリケーションで要求される高度な共有機能(たとえば顧客がフォームのフィールドに入力することは許すが他のページに移ることを禁止するなど)が可能となるような、ブラウザ共有方法及びシステムを提供することである。また別の課題は、ブラウザ共有に要するデータの転送量が少なく、家庭などのエンドユーザ環境におけるような、細いバンド幅においても良好なレスポンスが得られる、リアルタイムのブラウザ共有方法及びシステムを提供することである。

【0005】

【課題を解決するための手段】上記課題を解決するために、本発明は以下のように構成される。まず、本来のページを保持するウェブサーバから共有用のページを蓄積するためのコラボレーションサーバ(Collaboration Server)を用意する。コラボレーションサーバ(以下単にサーバと記す)はもとのウェブサーバで動的に生成されるページを共有用に蓄積するキャッシュマネージャ(Caching

4

Manager)と、各ユーザマシン上のブラウザを管理するノードマネージャ(NodeManager)間の通信と参加・退出などのセッション管理をするコミュニケーションマネージャ(Communication Manager)と、ウェブサーバ共有を実現する機構(ノードマネージャおよびページマネージャ)をダウンロードするための通常のウェブサーバと、各ページにページを管理するページマネージャを埋め込むエンベッダー(Embedder)からなる。

【0006】共有が行われる複数のユーザマシンは、Java および Script を実行することのできる既存のウェブブラウザと、各ページに埋め込まれたページマネージャ(PageManager)から構成される。ページマネージャは、ページコントローラおよびページコミュニケータの2つの部分からなる。ページコントローラ(PageController)は、ページ要素の変化を検出して、それをページコミュニケータ経由でほかのマシンに伝え、また他のマシンから受け取ったときには同じ変化を自ページの要素に対して起こさせる機能を持つ。ページの要素の変化とは、ページの移動、フォームの要素となるテキストやボタンの値の変化、ページのスクロール位置の変化、リモートポインタの操作などを指す。ページコミュニケータ(PageCommunicator)は、ノードマネージャとページコントローラの間の通信をつかさどる。ブラウザを管理するノードマネージャは、各ブラウザプロセスごとに用意され、各ページマネージャとサーバとの間の通信をつかさどる。なお上記サーバのハードウェア構成にコラボレーションのための特有なものは存在しない。ユーザマシンとサーバは名前が異なるだけであり、たとえば各ユーザマシンとサーバがまったく同一のハードウェアで構成されていても何ら構わない。

【0007】図2に本発明の全体構成図を示す。コラボレーションサーバはもとのウェブサーバで動的に生成されるページを共有用に蓄積するキャッシュマネージャと、ウェブサーバ共有を実現する機構をダウンロードするための通常のウェブサーバ(httpd)と、各ユーザマシン上のノードマネージャ間のセッション管理をするコミュニケーションマネージャと、ページマネージャを埋め込むエンベッダーからなる。複数のコンピュータ(各ユーザマシン)について共有のための機構は、ブラウザー(ウェブブラウザ1またはウェブブラウザ2)のプロセスごとに管理するモジュール(ノードマネージャ)と各ページごとに管理するモジュール(ページマネージャ)の2つのコンポーネントである。ページマネージャは、各ページ内でのページ要素ごとの状態を監視および変化を検出し、遠隔で対応するページマネージャとの間で情報を交換して、同じ状態になるように各ページ要素の設定を動的におこなう。またページマネージャは、ネストしたフレーム構造をもつウィンドウでの同期のために、フレームの階層構造(何番目のネストの何番目の位置)を調べ、これをIDとして対応するページマネージャと通信を行う。この

階層構造情報は、任意のブラウザにおいてクロスフレームセキュリティ機構による制限を受けずに取得できる。なお図2ではユーザマシンは2台であるが、3台以上も同様にブラウザ共有が可能である。

【0008】ブラウザを管理するノードマネージャは、各ページマネージャとサーバとの間の通信（セッション、同期）を行なう。ノードマネージャは、共有されたウェブウィンドウとは独立の移動しないページ内に常駐し、ページの移動ごとに動的に生成・消滅するページマネージャ同士の通信を制御する。また、ヒストリーなどのページにまたがった情報を管理する。ページマネージャとノードマネージャは、同一のドメインを持つ Java アプレットとして埋め込む。これにより、ページマネージャが埋め込まれるもとのページのドメインに関係なく、ページマネージャとノードマネージャの間で共有メモリーによるデータ通信が、任意のブラウザにおいてクロスフレームセキュリティ機構による制限を受けずに行われる。

【0009】このように構成することにより、既存のHTMLページに、その本来の構造を変えることなく、ページ共有のための機構を、既存ウェブサーバとブラウザの間で埋め込むことができるので、リアルタイムのウェブブラウザ共有が可能になる。

【0010】

【発明の実施の形態】以下にブラウザ共有システムの動作を図3のブラウザ内部の動作に基づき詳細に説明する。

【0011】1. ノードマネージャの始動

ノードマネージャは、顧客が共有開始となるページでログインしたとき、あるいはページ内に置かれた共有開始ボタンを押すことによって、新しいブラウザウィンドウが開かれ、そこにロードされる。このウィンドウは、ユーザから見て共有されるウィンドウとは独立に存在し、共有セッション中閉じられることはない。このウィンドウには、リモートポインターと通常モードの切り替えを行うためのユーザーインターフェイスなどが表示されている。このノードマネージャとサーバのコンポーネントであるコミュニケーションマネージャとの間で Java のソケットが張られ、共有データが複数のユーザマシン間で交換される準備ができる。さらに、ノードマネージャ・アプレットのクラス変数として、共有メモリーがアロケートされる。この中に、ページマネージャ との間のメッセージのキューが構成される

2. ページマネージャの共有ページへの埋め込み

ページマネージャは、各ページ要素の制御を行うスクリプトから成る ページコントローラ と、メッセージキューを通して ノードマネージャ と通信を行うアプレットで作られた ページコミュニケータ から構成される。サーバのエンベッダモジュールによって、この2つのモジュールが埋め込まれる。実際のコード例としては、HTMLページ内の文字列を読み込みながら、<BODY>タグ ので

たときにその直後に以下のコードを埋め込む。

```
【0012】<SCRIPT SRC="http://collabsvr/pagemgr.js"></SCRIPT>
<APPLET CODE="http://collabsvr/pagemgr.class"></APPLET>
```

【0013】ページコミュニケータとノードマネージャは同一のドメインを持つので、埋め込まれる側のページのドメインに関係なくセキュリティにふれずにデータを交換できる。これにより、複数ドメインのページからなるマルチフレームページが共有可能となる。

【0014】3. ページマネージャによる共有ページ内ページ要素の変化の検出

ページコントローラは、以下の要素に対するローカルでの変化の検出をおこない、その変化をリモートで対応するページコントローラに伝えて、ページの同期を実現する。

3-1. ページ移動の同期メカニズム

ページの変化は以下の動作に伴い起こる。

【0015】(1) ユーザ操作による、ページ内リンク上でのクリック

(2) ユーザ操作による、ブラウザーメニューでの操作 (Back/Forward ボタンおよび bookmark 選択)

(3) ページ内の記述による自動ロード (META 構文による指定およびアプレットやスクリプトによる記述)

【0016】このうち、(1) は次に述べるフォーム要素の変更イベント検出と同様にクリックイベントを検出してそれを伝えることによって同期する。(2) および (3) のケースでは、現在のページ上にある ページコントローラ が移動要求イベントを検出することができないので、新しくロードされたページ内の ページコントローラ が新しい URL を他のノードに伝える。ここで図4にネストしたフレームにおけるページ移動の処理例を示す。図4のステップ1にはブラウザ1および2の初期状態が図示されている。図4では、以下の初期状態（同期状態）が最終状態（同期状態）に遷移するケース（フレームセット2がフレームセット3へジャンプした場合）について図示している。

【0017】初期状態

FRAMESET [S1]

FRAME [F1]

FRAMESET [S2]

FRAME [F2]

FRAME [F3]

FRAME [F4]

【0018】最終状態

FRAMESET [S1]

FRAME [F1]

FRAMESET [S3]

FRAME [F5]

FRAME [F6]

【0019】なおここでS1、S2はフレームセット1、フレームセット2の、F1、F2、F3、F4、F5、F6はそれぞれフレーム1、2、3、4、5、6の略である。フレームセットの移動は、その第一フレームのページ(さらにネストして、第一要素がやはりフレームセットの場合には、順に下にたどっていく)に含まれるページマネージャによって代行される。

【0020】図4のステップ2では、フレームセット2がフレームセット3へのジャンプに伴い、フレーム2、3、4のページマネージャがまず破棄され、関連するメッセージキューが開放される。

【0021】図4のステップ3では、フレーム5、6がロードされる。関連するメッセージキューが配置され、フレーム5のページマネージャが”フレームセット3をロード”のメッセージを送信する。

【0022】図4のステップ4では、ブラウザ2のノードマネージャが同一のフレーム階層のメッセージキューへメッセージを配信し、フレーム2のページマネージャがフレームセット2のURLをロードする。そして同期状態に戻る。

【0023】3-2. フォーム要素の変更イベント
テキストフィールドやボタンなどのフォーム要素の同期に関しては、その性質に応じて2通りの方法で変化を検出する。

(1) ユーザ操作によるイベントを、イベントハンドラーのフックによって検出する

(2) 変更時に必ずしもイベントを発生しない要素に関して、タイマーによるポーリングによって検出する

(2)の方法は従来技術であるので説明を省略する。(1)の場合、ボタンなどのフォーム要素に対するユーザの操作はクリックイベントが発生するので、onclickのイベントハンドラーに対してページコントローラが自分用のハンドラーをセットすることによって、いわゆるイベントのフックをおこなわせることが可能である。しかし、一般にボタンなどの要素にはすでにユーザ定義のイベントハンドラがJavaScriptやVisualBasic Scriptによって記述されていることがあり、この場合イベントハンドラーをページコントローラが単純に置き換えただけでは、もとのハンドラーが処理されず、既存のページがもともとの記述どおりに動作しなくなる。そこでページロード時にもとのハンドラーを書き換えて、フック用のハンドラーに飛ばす行を書き加えるようにする。このようにすることにより、もとのハンドラーに影響を与えずに、ページコントローラがイベントをフックすることが可能となる。

【0024】3-3. ページのスクロール

スクロール時に発生するイベント(onScroll)をフックすることによって簡単に検出できるが、ドラッグの場合イベントが発生しつづけて非常にデータ通信量が多くなってしまうので、タイマーによるポーリングでの位置検

出を行う。

【0025】3-4. リモートポインターの操作

リモートポインタは、たとえばマイクロソフト社のインターネットエクスプローラの場合、以下のようにDIVエレメントとして付加することにより、任意ページに付け加えることができる。ネットスケープ社のネットスケープコミュニケーターの場合も同様の手法で行なうことができる。

【0026】var str = '<DIV style="position:absolute; overflow:none; width=50px height=50px"></DIV>';document.body.insertAdjacentHTML("AfterBegin", str);

【0027】ポインターの移動は、windowに対するmoveMove イベントによって対応する位置にDIVを移動させることによって行われる

【0028】図5に、本発明の方法によりブラウザ共有が行えるようになるまでの処理をフローチャートによって説明する。利用者が共有開始のページでログインしたとき、あるいはページ内に配置された共有開始ボタンをおすことによって、共有の処理が開始される。ステップ510では、新しいブラウザウィンドウを開き、そこにノードマネージャをロードする。このノードマネージャのウィンドウは、共有されるブラウザウィンドウとは独立に存在し、共有セッション中に閉じられることはないものとする。このノードマネージャのウィンドウには、リモートポインタと通常モードとの切り替えを行うユーザインタフェースなどを表示してもよい。ステップ520では、コラボレーションサーバのコンポーネントであるコミュニケーションマネージャとノードマネージャとの間をJavaのソケットによって接続する。このソケットを用いて、共有するための情報をノード間で交換することになる。ステップ530では、サーバのエンベッドモジュールによって、HTMLページにページマネージャを埋め込む。ただし、ページマネージャは、各ページ要素の制御を行うスクリプトからなるページコントローラと、メッセージキューを通してノードマネージャと通信を行うアプレットからなるページコミュニケーターから構成される。ページコミュニケーターとノードマネージャは、同一ドメインであるので、埋め込まれるHTMLページのドメインに関係なく、セキュリティの問題を避けてデータを交換することができる。このことにより、複数のドメインのページからなるマルチフレームページの共有が可能となる。以上でブラウザ共有が行えるようになるまでの処理は終了である。

【0029】図6に、本発明の方法によりブラウザ共有が行えるようになった後、利用者のブラウザの操作を共有するための処理をフローチャートによって説明する。ただし、フローチャートで記述する制約のため実際には、イベントが発生したときに、そのイベントに対して設定されているハンドラが起動される場合にも、判断部

で代用している。また、ハンドラによらない処理でもタイマを設定することにより、定期的に処理を行っているが、フローチャート上では単なるループで代用している。ステップ610で、ブラウザがページマネージャを埋め込まれたHTMLページをロードする。次にステップ620で、ノードマネージャのアプレットのクラス変数として、共有メモリを割り当てる。この中にページマネージャとの間のメッセージのキューを作成する。また、onclickのイベントハンドラに対して、ページコントローラが自分用のハンドラを設定する。

【0030】さらに、その他のイベントについても必要に応じて同様の処理を行う。ステップ630では、終了判定を行なう。共有終了ボタンが押されるなどすると、共有が終了される。ステップ640では、ページ移動判定を行なう。ページの移動は、利用者によるページ内リンクでのクリックによる場合は、現在のページコントローラがクリックのイベントを検出することにより判定する。ブラウザメニューでの操作（戻る、進む、ブックマーク）や、ページ内の記述による（META構文、アプレット、スクリプト）などによって発生する移動は、新しくロードされたページ内のページコントローラがブラウザウィンドウのlocation属性を参照することにより判定する。

【0031】ステップ642ではページ移動処理を行なう。利用者によるブラウザメニューでの操作及びページ内の記述によるページ移動のときは、新しくロードされたページ内のページコントローラが新しいURLを他のノードに伝える。ステップ644ではページのアンロードを行なう。ページ移動に伴い、ブラウザが現在のページをアンロードする。そしてステップ646では、ページマネージャを終了する。現在のページをアンロードする過程でこのページ上のページマネージャが終了する。このとき、ページマネージャとの間のメッセージのキューを削除し、この共有メモリを解放する。

【0032】ステップ650ではフォーム判定を行なう。テキストフィールドやボタンなどのフォーム要素の同期については、フォーム要素の性質に応じて2通りの方法で判定する。利用者の操作によるイベントをイベントハンドラをフックすることにより判定する。変更時に必ずしもイベントを発生しない要素の変更は、タイマで定期的に値を確認することにより判定する。ステップ655では、フォーム処理を行なう。利用者の操作がイベントを発生させる種類の操作である場合には、ステップ642のページ移動処理と同様の処理を行う。変更時に必ずしもイベントを発生しない要素の変更の場合には、変更したフォーム要素の値を他のノードに送信する。処理はその後、ステップ630へ戻る。ステップ660では、スクロール判定を行なう。タイマにより定期的に位置を検出することにより、スクロールの操作を判定する。ステップ665では、スクロール処理を行なう。新

しいスクロールの位置を他のノードに送信する。処理はその後、ステップ630へ戻る。ステップ670ではポインタ判定を行なう。ウィンドウに対するmouseMoveイベントにより判定する。ステップ685ではポインタ処理を行なう。新しいポインタの位置を他のノードに送信する。ポインタは、DIVエレメントとして任意のHTMLページに以下のように付加することができる。

```
【0033】 var str = '<DIV style="position: absolute; overflow: none; width: 50px; height: 50px"><IMG src = "pointer.gif"></DIV>'; document.body.insertAdjacentHTML("AfterBegin", str);
```

【0034】ポインタの移動は、mouseMoveイベントから得られた位置にDIVを移動することによって行う。処理はその後ステップ630へ戻る。ステップ680では受信判定を行なう。ページコミュニケータがメッセージキューを確認することにより、他のノードからのメッセージの受信があるかどうかを判定する。ステップ685では、受信処理を行なう。受信したメッセージの内容により、適切な処理を行う。利用者によるブラウザメニューでの操作及びページ内の記述によるページ移動を指示するメッセージを受信したときは、ウィンドウのlocationに受信したURLを設定し、同じHTMLページをロードする。フォーム要素の値を変更するメッセージを受信したときは、指定されたようにフォームの要素を変更する。スクロールの位置を変更するメッセージを受信したときは、指定されたようにスクロールの位置を変更する。ポインタの位置を変更するメッセージを受信したときは、指定されたようにポインタの位置を変更する。

【0035】図7には、本発明において使用されるサーバ、複数のコンピュータ（ユーザマシン）のハードウェア構成の一実施例を示す。システム100は、中央処理装置（CPU）1とメモリ4とを含んでいる。CPU1とメモリ4は、バス2を介して、補助記憶装置としてのハードディスク装置13（またはMO、CD-ROM23、DVD等の記憶媒体駆動装置）とIDEコントローラ25を介して接続してある。同様にCPU1とメモリ4は、バス2を介して、補助記憶装置としてのハードディスク装置30（またはMO28、CD-ROM23、DVD等の記憶媒体駆動装置）とSCSIコントローラ27を介して接続してある。フロッピーディスク装置20はフロッピーディスクコントローラ19を介してバス2へ接続されている。

【0036】フロッピーディスク装置20には、フロッピーディスクが挿入され、このフロッピーディスク等やハードディスク装置13（またはMO、CD-ROM、DVD等の記憶媒体）、ROM14には、オペレーティングシステムと協働してCPU等に命令を与え、本発明を実施するためのコンピュータ・プログラムのコード若しくはデータを記録することができ、メモリ4にロードされることによって実行される。このコンピュータ・プ

プログラムのコードは圧縮し、または、複数の分割して、複数の媒体に記録することもできる。

【0037】システム100は更に、ユーザ・インターフェース・ハードウェアを備え、入力をするためのポインティング・デバイス（マウス、ジョイスティック等）7またはキーボード6や、視覚データをユーザに提示するためのディスプレイ12を有することができる。またパラレルポート16を介してプリンタを接続することや、シリアルポート15を介してモデムを接続することが可能である。このシステム100は、シリアルポート15およびモデムまたは通信アダプタ18（イーサネットやトークンリング・カード）等を介してネットワークに接続し、他のウェブサーバ、他のコンピュータ等と通信を行う。またシリアルポート15若しくはパラレルポート16に、遠隔送受信機器を接続して、赤外線若しくは電波によりデータの送受信を行うことも可能である。

【0038】スピーカ23は、オーディオ・コントローラ21によってD/A（デジタル/アナログ変換）変換された音声信号を、アンプ22を介して受領し、音声として出力する。また、オーディオ・コントローラ21は、マイクロフォン24から受領した音声情報をA/D（アナログ/デジタル）変換し、システム外部の音声情報をシステムにとり込むことを可能にしている。

【0039】このように、本発明のサーバ、複数のコンピュータは、通常のパーソナルコンピュータ（PC）やワークステーション、ノートブックPC、パームトップPC、ネットワークコンピュータ、コンピュータを内蔵したテレビ等の各種家電製品、通信機能を有するゲーム機、電話、FAX、携帯電話、PHS、電子手帳、等を含む通信機能を有する通信端末、または、これらの組合せ

10

20

30

によって実施可能であることを容易に理解できるであろう。ただし、これらの構成要素は例示であり、その全ての構成要素が本発明の必須の構成要素となるわけではない。

【0040】

【発明の効果】本発明により、従来技術では実現できなかった多人数間での高度なリアルタイムのブラウザ共有機能を実現される。さらにページ移動の同期、フォーム要素の入力操作の同期、スクロール操作の同期、遠隔ポインタの同期、アノテーションの同期が可能であり、クライアントマシンにはJavaおよびスクリプトの機能を有する既存のブラウザだけでよい。そして外付けのプログラムやインストールするモジュール・プラグイン等を必要としないので、ユーザ側にインストールの負担をかけない、同期のために必要とするデータ通信量が少ない、ブラウザ共有システムが提供される。

【図面の簡単な説明】

【図1】従来のブラウザ共有例を示す図である。

【図2】本発明の全体構成を示す図である。

【図3】本発明のブラウザ内部の動作を示す図である。

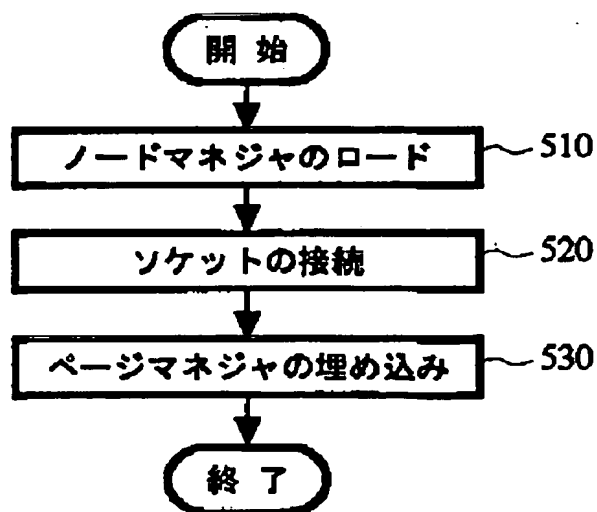
【図4】ネストしたフレームにおけるページ移動の処理例を示す図である。

【図5】本発明の方法によりブラウザ共有が行えるようになるまでのフローチャートである。

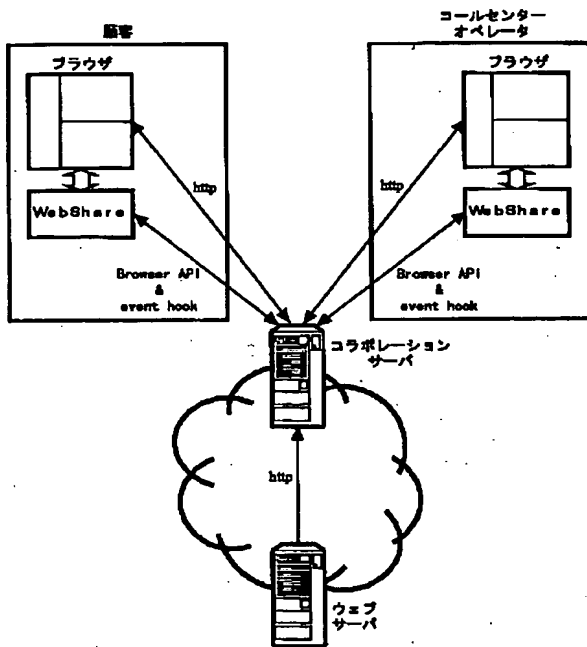
【図6】本発明の方法によりブラウザ共有が行えるようになった後、利用者のブラウザの操作を共有するためのフローチャートである。

【図7】本発明において使用されるサーバ、複数のコンピュータのハードウェア構成例を示す図である。

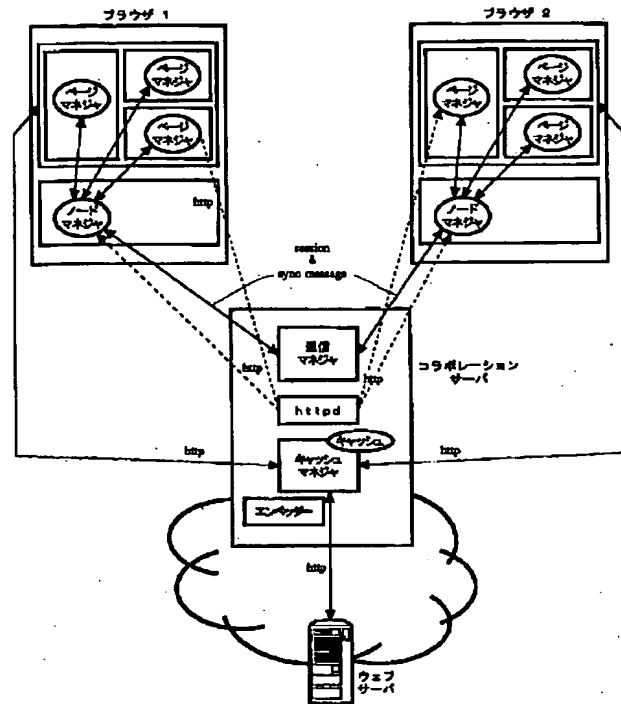
【図5】



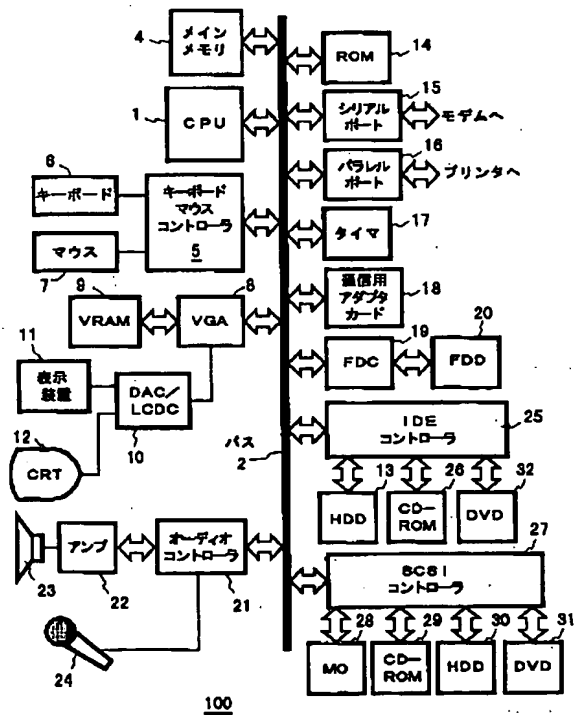
【図 1】



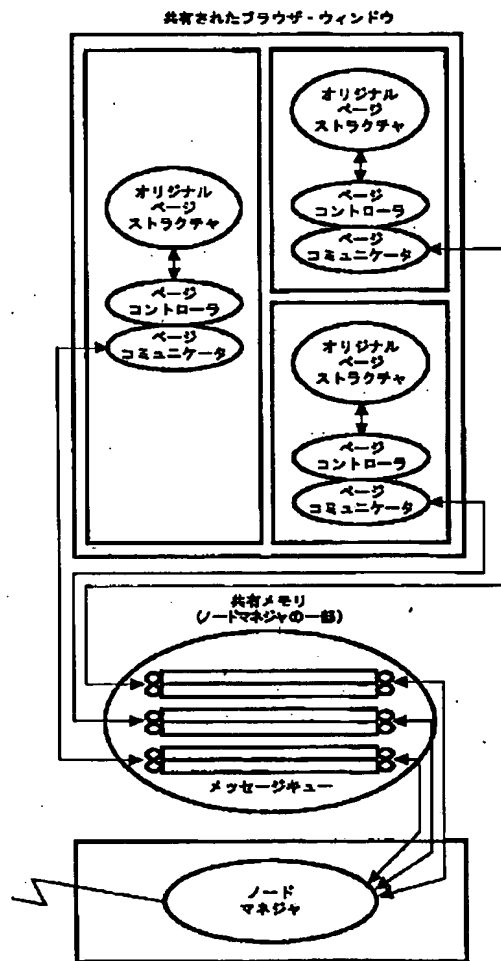
【図 2】



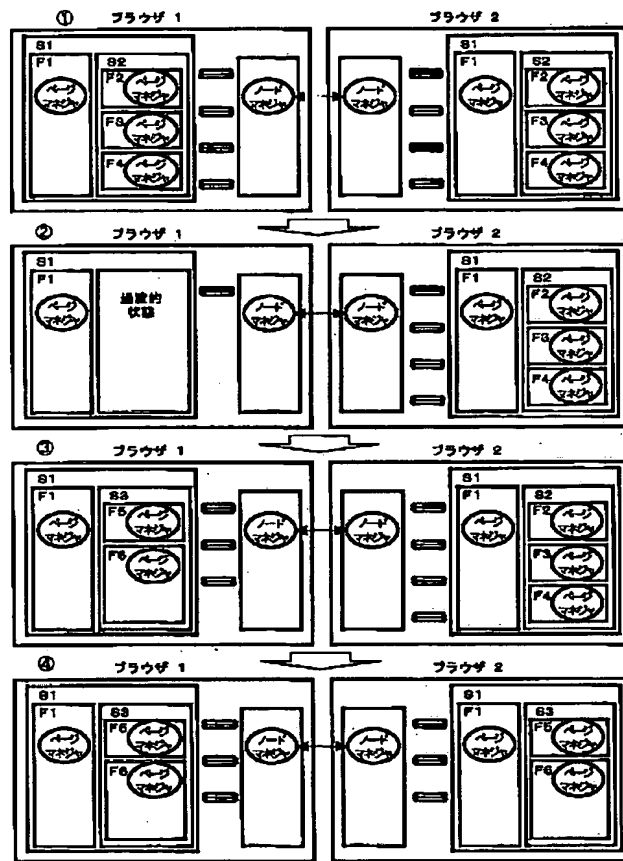
【図 7】



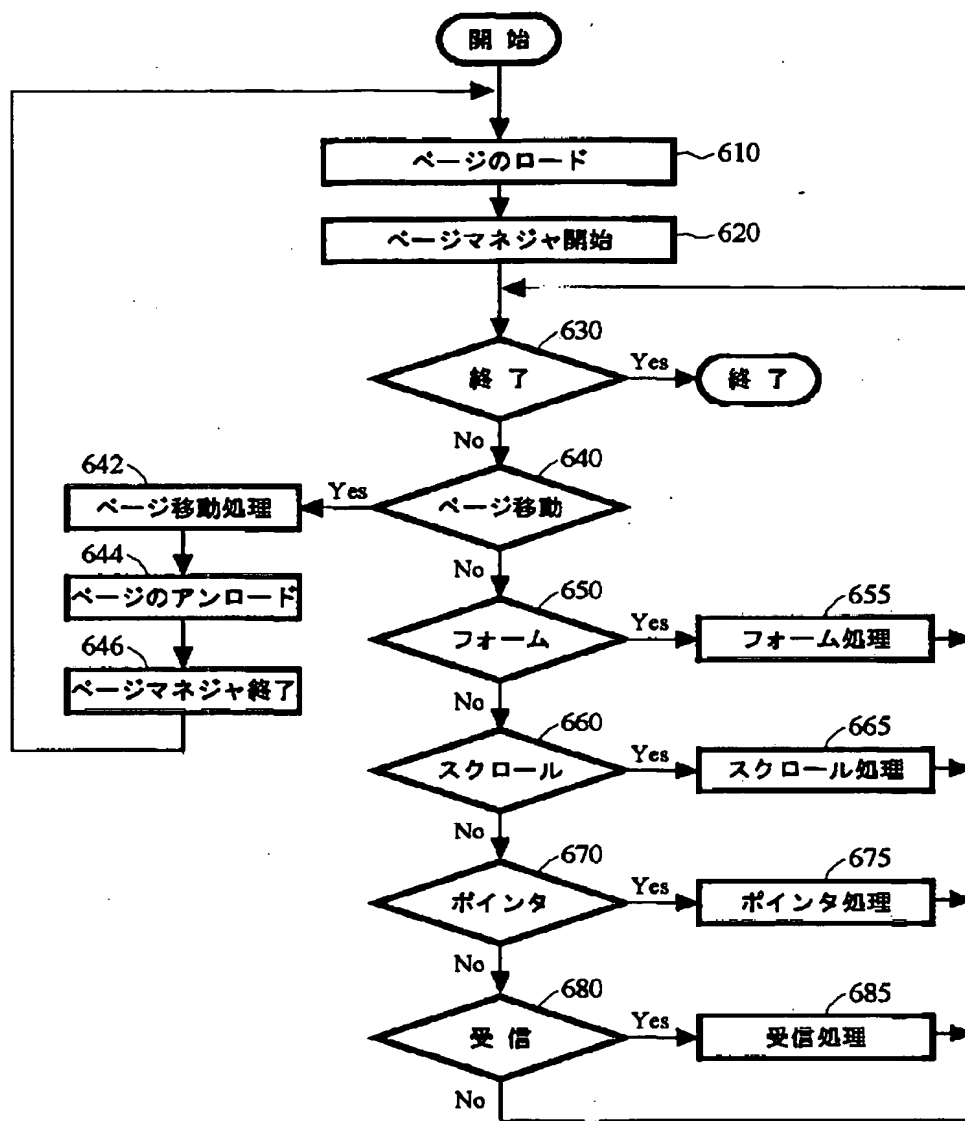
【図3】



【図4】



【図6】



フロントページの続き

(72)発明者 小林 真
 神奈川県大和市下鶴間1623番地14 日本ア
 イ・ビー・エム株式会社 東京基礎研究所
 内
 (72)発明者 篠▲崎▼ 雅英
 神奈川県大和市下鶴間1623番地14 日本ア
 イ・ビー・エム株式会社 東京基礎研究所
 内

(72)発明者 坂入 隆
 神奈川県大和市下鶴間1623番地14 日本ア
 イ・ビー・エム株式会社 東京基礎研究所
 内
 Fターム(参考) 5B045 BB32 DD12
 5B075 KK12 PQ02 PQ42
 5B082 FA12 FA16 GB06 HA08
 5B085 BE07 BG07

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER: _____**

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.